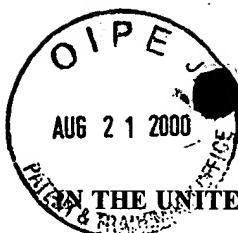


AT9-97-206



PATENT

178/6AM-2771#  
K. Ward  
8/29/00  
#13 Appeal Brief  
(3)

In re application of: I. Holland et al.

Serial No.: 08/890,643

Art Unit: 2771

Filed: July 9, 1997

Examiner: J. Mills

For: SIMULATION OF MEMORY-MAPPED I/O

Box AF

Assistant Commissioner for Patents

Washington, D.C. 20231

**TRANSMITTAL OF APPEAL BRIEF  
(PATENT APPLICATION - 37 CFR 1.192)**

**1. Transmitted herewith in triplicate is the APPEAL BRIEF in this application with respect to the Notice of Appeal filed on June 19, 2000.**

*NOTE: "The appellant shall, within 2 months from the date of the notice of appeal under § 1.191 in an application, reissue application, or patent under reexamination, or within the time allowed for response to the action appealed from, if such time is later, file a brief in triplicate." 37 CFR 1.192(a) (emphasis added).*

**2. STATUS OF APPLICANT**

This application is on behalf of

- ☒ other than a small entity
- ☐ small entity
- verified statement:
  - ☐ attached
  - ☐ already filed

RECEIVED  
AUG 23 2000  
TC 2700 MAIL ROOM

**3. FEE FOR FILING APPEAL BRIEF**

Pursuant to 37 CFR 1.17(f) the fee for filing the Appeal Brief is:

- ☐ small entity \$150.00
- ☒ other than a small entity \$300.00

**Appeal Brief fee due \$300.00**

**CERTIFICATE OF MAILING (37 CFR 1.8(a))**

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to Box AF, Assistant Commissioner for Patents, Washington, D.C. 20231.

Date: 8-15-00

Toni Asendorf

(Type or print name of person mailing paper)

Toni Asendorf  
(Signature of person mailing paper)

08/22/2000 HAMMOND 00000056 090447 08890643

01 FC:120

300.00 CH

**4. EXTENSION OF TERM**

**NOTE:** The time periods set forth in 37 CFR 1.192(a) are subject to the provision of § 1.136 for patent applications. 37 CFR 1.191(d). Also see Notice of November 5, 1985 (1060 O.G. 27).

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136 apply.

(complete (a) or (b) as applicable)

- (a) ☐ Applicants petition for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

Extension (months)	Fee for other than small entity	Fee for small entity
<input type="checkbox"/> one month	\$ 110.00	\$ 55.00
<input type="checkbox"/> two months	\$ 380.00	\$190.00
<input type="checkbox"/> three months	\$ 870.00	\$435.00
<input type="checkbox"/> four months	\$1,360.00	\$680.00
Fee		\$ _____

If an additional extension of time is required, please consider this a petition therefor.

(check and complete the next item, if applicable)

- ☐ An extension for \_\_\_\_\_ months has already been secured and the fee paid therefor of \$ \_\_\_\_\_ is deducted from the total fee due for the total months of extension now requested.

Extension fee due with this request \$ \_\_\_\_\_

or

- (b) ☒ Applicants believe that no extension of term is required. However, this conditional petition is being made to provide for the possibility that applicants have inadvertently overlooked the need for a petition and fee for extension of time.

**5. TOTAL FEE DUE**

The total fee due is:

Appeal Brief fee \$300.00

Extension fee (if any) \$ \_\_\_\_\_

**TOTAL FEE DUE \$300.00**

**6. FEE PAYMENT**

- ☐ Attached is a check in the sum of \$ \_\_\_\_\_
- ☒ Charge Account No. 09-0447 (AT9-97-206) the sum of \$300.00.

A duplicate of this transmittal is attached.

**7. FEE DEFICIENCY**

**NOTE:** If there is a fee deficiency and there is no authorization to charge an account, additional fees are necessary to cover the additional time consumed in making up the original deficiency. If the maximum, six-month period has expired before the deficiency is noted and corrected, the application is held abandoned. In those instances where authorization to charge is included, processing delays are encountered in returning the papers to the PTO Finance Branch in order to apply these charges prior to action on the cases. Authorization to charge the deposit account for any fee deficiency should be checked. See the Notice of April 7, 1986; 1065 O.G. 31-33.

- ☒ If any additional extension and/or fee is required, this is a request therefor and to charge Account No. 09-0447 (AT9-97-206).

AND/OR

- ☒ If any additional fee for claims is required, charge Account No. 09-0447 (AT9-97-206).

Reg. No.: 36,571

  
SIGNATURE OF ATTORNEY

Tel. No.: (512) 370-2851

Kelly K. Kordzik  
WINSTEAD SECHREST & MINICK P.C.  
5400 Renaissance Tower  
1201 Elm Street  
Dallas, Texas 75270

::ODMA\PCDOCS\AUSTIN\_1\146402\1  
207:7047-P170X1

AT9-97-206



PATENT

- 1 -

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	:	Before the Examiner:
I. Holland et al.	:	J. Mills
Serial No.: 08/890,643	:	Group Art Unit: 2771
Filed: July 9, 1997	:	Intellectual Property Law
	:	IBM Corp.
Title: SIMULATION OF	:	11400 Burnet Road
MEMORY-MAPPED I/O	:	Austin, TX 78758

August 15, 2000

**APPEAL BRIEF**

Box AF  
Assistant Commissioner for Patents  
Washington, D. C. 20231

RECEIVED  
AUG 23 2000  
TC 2700 MAIL ROOM

I. **REAL PARTY IN INTEREST**

The real party in interest is International Business Machines Corp., which is the assignee of the entire right, title and interest in the above identified patent application.

---

**CERTIFICATION UNDER 37 C.F.R. § 1.8**

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Box AF, Assistant Commissioner for Patents, Washington, D.C. 20231, on August 15, 2000.

Signature

Toni Asendorf

(Printed name of person certifying)

## II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to appellant, appellant's legal representative, or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

## III. STATUS OF CLAIMS

Claims 1-27 are pending in the Application. Claims 1-27 stand rejected.

## IV. STATUS OF AMENDMENTS

No amendments were filed by Applicants subsequent to receipt of the Final Rejection (Paper No. 11).

## V. SUMMARY OF INVENTION

A memory-mapped file operation begins with a read only file 102 stored within a data processing system storage media, such as a hard disk 101. See Figure 1; Specification, page 6, lines 15-17. Memory-mapped I/O provides a memory representation of the read only file 102 within the virtual memory 103 associated with an application, which includes the RAM associated with the processor. Specification, page 6, lines 17-19. The memory representation of the read only file 102 is represented as 104 within the virtual memory 103. Specification, page 6, lines 19-21.

To create the representation in the memory 104, the operating file system determines how large is the read only file 102, then allocates a block size in the memory 103 for the number of pages that the read only file 102 is going to reside in. Specification, page 7, lines 1-4. The software application 105 is then given a reference pointer to memory, which is in contrast to a standard file I/O API operation (see Figure 4) whereby the operating system reads the file into a buffer from the hard disk 101. Specification, page 7, lines 4-6.

The result of this process is that any modifications made to the read only file in memory 104 are reflected back onto the read only file 102 on the hard disk 101 (this is applicable only for read/write files), a page is faulted into the memory block 104 when the page is touched, or referenced, and the operating system then reads the exact size for that page out of the file and maps it into the memory 104 at that location. Specification, page 7, lines 7-12.

Figures 2 and 3 illustrate the process of the present invention whereby memory mapping is done via the system loader. Specification, page 7, lines 17-18. What is desired is for a block of data, such as data within a database, to be memory-mapped into the virtual memory 103 as the representation in memory 203, so that the software application 105 can merely use a reference pointer to the memory 203 instead of reading a file into a buffer from the hard disk 101, which is a less efficient method especially with very large data files. Specification, page 7, lines 18-page 8, line 2.

Referring to the Specification, page 8, lines 3-21, in order to perform this process in certain operating systems, such as OS/2, Windows 3.X, etc., the data needs to be converted into an executable file, represented in FIGURE 2 as mydata.so/dll 201. Executable files generally include code and accompanying data. In this case, the data portion is optional. The process is performed by taking a read only file 301 and putting it through a code converter 302, which is a tool that wrappers the read only data with code headers and records in order to make the operating system loader 202 believe that the read only file is executable code. The system loader 202 then believes that the read only file is executable code and performs a memory-map operation to memory-map the read only file as the image representation in memory 203. The software application 105 is then given a reference pointer to the memory 203 in order to access the read only file.

The result of the foregoing operation is that the read only file is converted into an executable/shared library file 305, such as a DLL executable. The system loader 202 looks

at the read only file as just another DLL and maps it right into memory 103. An option is for the R/O file to be passed through a tool in steps 303-304 that generates an object file and then generates an executable or shared libraries using the object file as an input into a link editor or generates the executable or shared library directly skipping the intermediate step.

#### VI. ISSUE

Are claims 1-27 properly rejected under 35 U.S.C. § 102(e) as being anticipated by *Blackard et al.* (U.S. Patent No. 5,301,302)?

#### VII. GROUPING OF CLAIMS

Claims 1-3, 7-9, and 13-15 are to be considered as a group. Claims 4, 10, and 16 are to be considered as a group. Claims 5, 11, and 17 are to be considered as a group. Claims 6, 12, and 18 are to be considered as a group. Claim 19 is to be considered separately from the other claims. Claim 20 is to be considered separately from the other claims. Claims 21-23 are to be considered as a group. Claims 24-26 are to be considered as a group. Claim 27 is to be considered separately from the other claims. The reasons why the foregoing claim groups have been designated are provided in Section VIII below, wherein each of the noted grouping of claims are separately argued by Applicants for the reasons given.

#### VIII. ARGUMENT

Claims 1-27 are not properly rejected by 35 U.S.C. § 102(e) as being anticipated by *Blackard et al.*

As the Examiner is well aware, for a claim to be anticipated, each and every element of the claim must be found within the single prior art reference. Claim 1 specifically recites that a read-only file is converted into an executable file. This is not performed by *Blackard*. The Examiner supports his rejection by referring to column 8, line 21, *et seq.* in *Blackard*.

However, *Blackard* discloses therein that a simulator 10 copies BIOS 13 from ROM 15 into the operating system's shared memory segment 16. The BIOS 13 is then used to load the DOS for which the application was originally written. The simulator then translates and executes that operating system. This does not disclose the conversion of a read-only file into an executable file. Merely instead, BIOS is copied from ROM into the shared memory segment 16.

Furthermore, the Examiner then asserts that the memory-mapping step of Claim 1 is taught in *Blackard* in column 16, line 21, *et seq.* This portion of *Blackard* refers to translating the addresses of a first processing system into the addresses of a second processing system. Column 16, lines 16-17. To do this, the memory of the first processing system is mapped into the memory of the second processing system. Column 16, lines 18-20. This portion of *Blackard* is not referring to the BIOS copied from ROM into the shared memory segment 16, as is disclosed in column 8, lines 21, *et seq.* Therefore, *Blackard* does not teach that the BIOS copied from ROM is then memory-mapped. The Examiner is taking two separate and unrelated teachings within *Blackard* and combining them to assert that *Blackard* anticipates Claim 1. This is not permissible. To anticipate, the elements taught in the prior art must be arranged as required by the claim. *In re Bond*, 910 F.2d 831, 15 U.S.P.Q.2d 1566 (Fed. Cir. 1990). Furthermore, the identical invention must be shown in as complete detail as is contained in the claim. *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 U.S.P.Q.2d 1913, 1920 (Fed. Cir. 1989); *see also* MPEP § 2131. Consequently, the case law does not support the Examiner's rejection, because the Examiner's citation of the language in column 8 and the language in column 16 in *Blackard* does not result in as complete detail as is contained within the claim.

Claims 7 and 13 are also not anticipated for the same reasons as given above with respect to Claim 1.



Claim 4 specifically recites that the converted file appears to the operating system loader as a shared library file. The Examiner asserts that the "shared memory is taught by the reference as the shared memory segments taught on column 16, line 21." Applicants respectfully traverse this assertion by the Examiner. First of all, the Examiner is referring to a "shared memory", while the claim specifically recites a "shared library file", which is not the same as a shared memory. Therefore, the Examiner has failed to prove a *prima facie* case of anticipation in rejecting Claim 4. Furthermore, the shared memory segments discussed in column 16 of *Blackard* merely refer to a memory that is used to store an image of the memory of another system, which is not the same as a shared library file. A library file is a collection of routines used by a processing system, and a shared library file is a collection of routines that can be used within a processing system by several applications. MICROSOFT PRESS COMPUTER DICTIONARY, p. 209, copyright 1991 by Microsoft Press. Therefore, Claim 4 is not anticipated by *Blackard*.

Claims 10 and 16 are also not anticipated for the same reason as given above for Claim 4.

With respect to Claim 5, the Examiner asserts that the read-only file being a database file is inherent in the teaching of the reference in that the data in the read-only file may be formatted as a database. Applicants respectfully traverse this assertion. The Examiner has already asserted that the read-only file is the BIOS 13 copied from ROM. BIOS is not a database file. For this reason alone, the Examiner has failed to prove that *Blackard* teaches that the read-only file converted into the executable file is a database file.

Furthermore, the Examiner's assertion that the recitation in Claim 5 is inherently taught in *Blackard* is an incorrect interpretation of the law. It is stated in MPEP § 2131 that a "claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. vs. Union Oil Co. of California*, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987).

In that case, it was asserted by one party that a prior art patent process did not anticipate an invention, because the heel did not function as a heat sink. However, the Court asserted that the property of the heel was to still function as a heat sink even though it was not identified specifically as a "heat sink" within the patent. Thus, when the case law refers to a prior art reference inherently teaching an element, it is referring to an element having the same properties as the device asserted to be anticipated by the prior art reference. That is not the case with the present application, and specifically Claim 5. The Examiner is asserting that the read-only file being a database file is inherent in the teaching of the reference in that the data in the read-only file may be formatted as a database. However, the Examiner has failed to point to any portion of *Blackard* that teaches that a read-only file may be formatted as a database, and more specifically, nowhere in *Blackard* has the Examiner pointed to that the BIOS copied from ROM may be a database file. Therefore, *Blackard* does not expressly or inherently teach that a read-only file to be converted into an executable file may be a database file.

Claims 11 and 17 are also not anticipated for the same reason as given with respect to Claim 5.

Claim 6 specifically recites that the converting step further comprises the step of wrapping the read-only file with executable code. The Examiner has failed to point to an express teaching in *Blackard* of this limitation. Instead, the Examiner has asserted that this limitation is inherent in *Blackard* in its discussion of instruction address translation in column 13, line 22, *et seq.* Applicants respectfully disagree. The instruction address translation discussed in *Blackard* does not teach the wrapping of anything with executable code. Instead, *Blackard* merely teaches that a new instruction pointer 31 and code segment 33 whose values are determined at runtime are converted into the simulator machine address of the corresponding translation for those instructions that transfer control dynamically. In other words, the code segments on the disk are at a different location than

the code segments in memory where it actually gets loaded. As a result, when the code segment gets loaded into memory, an address translation is performed to determine where it is located in memory. There is nothing within this portion of *Blackard*, or anywhere else within *Blackard*, that discusses or teaches the wrapping of a read-only file with executable code. Nothing within *Blackard* teaches either expressly or inherently an ability or property for wrapping a file with executable code. Therefore, Claim 6 is not anticipated by *Blackard*.

Claims 12 and 18 are also patentable over *Blackard* in view of the foregoing remarks with respect to Claim 6.

Claims 19 and 20 recite, respectively, that the read-only file is an image file and an audio file. Again, the Examiner has failed to point to any express teaching in *Blackard* whereby a read-only file is either an image file or an audio file. Instead, the Examiner merely asserts that "the type of data and type of file is inherent in the teaching of the reference and the data could be either image or audio." Applicants respectfully traverse this assertion by the Examiner because it is the Examiner's own opinion without any type of support whatsoever within *Blackard*. In asserting that this is inherent in the teaching of *Blackard* the Examiner has even failed to refer to any specific language within *Blackard*. If the Examiner asserts that there is an inherent teaching in the reference, the Examiner must still point to such a teaching. Applicants respectfully assert that the Examiner must specifically point to language within *Blackard* that either expressly or inherently teaches these limitations, or the Examiner's assertion of anticipation must fail.

Moreover, Applicants again refer to the Examiner's rejection of Claims 1, 7 and 13, whereby the Examiner refers to the language recited in column 8, line 21, *et seq.* in *Blackard* for teaching these claim limitations. The Examiner has asserted that the BIOS copied from ROM teaches the read-only file converted into an executable file. If this were to be true, then the Examiner's rejection of Claims 19 and 20 cannot stand, because it is quite clear that a BIOS cannot be an image file or an audio file.

Furthermore, with respect to each and every "inherency" argument presented by the Examiner above, Applicants respectfully assert that the Examiner must support such inherency arguments with facts and objective evidence to reasonably support his determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art. MPEP § 2112. The Examiner has further cited U.S. Patent Nos. 5,802,554 and 5,440,710 to support his assertion that read only files may be by design choice executable. In response, Applicants respectfully assert that the Examiner must show why one skilled in the art at the time the invention was made would have taken the teachings in these two cited patents and combined them with the teachings of *Blackard* to arrive at the claimed invention.

With respect to Claims 21-23, the Examiner asserts that the wrapping feature of wrapping read only code with code headers in order to cause the operating system to view the file as an executable file is inherent in the translation or conversion of the read only file for the simulator of the cited prior art patent. Applicants respectfully traverse this assertion by the Examiner. First of all, as required under MPEP § 2112, Applicants respectfully request that the Examiner must support such an inherency argument with objective evidence and not merely his own opinion. Furthermore, Applicants disagree with the Examiner. With respect to read only files, a header code is not used. Instead, when a read only file is established, a memory map allocates a chunk of memory for that file, backs that file onto disk, and returns a pointer to the memory location that is allocated for that file. Therefore, when that file is then touched or the pointer is touched, the page will be faulted in. Thus, code headers are not utilized to wrap read only code to make the file appear to an operating system loader as executable code.

With respect to Claims 24-27, the Examiner has merely stated that they are rejected in view of the analysis with respect to the previous Claims 1-23 and are rejected on that basis. Applicants respectfully assert that this is an insufficient basis for rejecting Claims

24-27. It is the Examiner's responsibility when showing a *prima facie* case of anticipation or obviousness to specifically address each and every limitation within a claim. Since Claims 24-27 recite limitations that are not found within Claims 1-23, the Examiner's analysis for rejecting Claims 1-23 is insufficient to support a *prima facie* case of anticipation in rejecting Claims 24-27. More specifically with respect to Claims 24-26, nowhere within *Blackard* is it taught or suggested that the disclosed processor operates in a protected mode. Since the Examiner has not in any way specifically addressed this claim limitation, Applicants respectfully assert that the Examiner has failed to prove a *prima facie* case of anticipation in rejecting Claims 24-26.

Specifically with respect to Claim 27, again the Examiner has failed to specifically address the claim limitations within Claim 27, and as a result, the Examiner has failed to prove a *prima facie* case of anticipation in rejecting Claim 27. Nowhere within *Blackard* is it taught or disclosed to convert a read only file into a shared library with an executable code segment, wherein the shared library includes a reference pointer to data residing in the executable code segment. Further, *Blackard* does not teach or disclose an application that needs to use the data referencing a pointer to the data, and an operating system faulting a page containing a memory segment wherein the read only file resides into memory.

IX. CONCLUSION

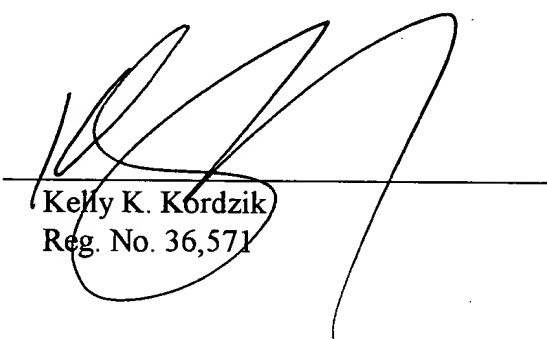
As a result of the foregoing, Applicants respectfully assert that the claims are allowable over *Blackard*.

Respectfully submitted,

WINSTEAD SECHREST & MINICK P.C.

Attorneys for Applicant

By:



Kelly K. Kordzik  
Reg. No. 36,571

5400 Renaissance Tower  
1201 Elm Street  
Dallas, Texas 75270  
(512) 370-2851

::ODMA\PCDOCS\AUSTIN\_1\146397\1  
207:7047-P170X1

**APPENDIX**

1. A method comprising the steps of:  
converting a read only file into an executable file; and  
memory-mapping the converted file into memory.
2. The method as recited in claim 1, wherein the memory is virtual memory associated with an application running in a processor.
3. The method as recited in claim 2, wherein the converted file is memory-mapped from system storage by an operating system loader.
4. The method as recited in claim 3, wherein the converted file appears to the operating system loader as a shared library file.
5. The method as recited in claim 3, wherein the read only file is a database file.
6. The method as recited in claim 3, wherein the converting step further comprises the step of wrapping the read only file with executable code.
7. A data processing system comprising:  
circuitry for converting a read only file into an executable file; and  
circuitry for memory-mapping the converted file into memory.

8. The data processing system as recited in claim 7, wherein the memory is virtual memory associated with an application running in a processor.

9. The data processing system as recited in claim 7, wherein the converted file is memory-mapped from system storage by an operating system loader.

10. The data processing system as recited in claim 9, wherein the converted file appears to the operating system loader as a shared library file.

11. The data processing system as recited in claim 7, wherein the read only file is a database file.

12. The data processing system as recited in claim 7, wherein the converting circuitry further comprises circuitry for wrapping the read only file with executable code.

13. A computer program tool adaptable for storage in a storage medium, comprising:

program code for converting a read only file into an executable file; and  
program code for memory-mapping the converted file into memory.

14. The computer program tool as recited in claim 13, wherein the memory is virtual memory associated with an application running in a processor.

15. The computer program tool as recited in claim 14, wherein the converted file is memory-mapped from system storage by an operating system loader.



16. The computer program tool as recited in claim 15, wherein the converted file appears to the operating system loader as a shared library file.

17. The computer program tool as recited in claim 15, wherein the read only file is a database file.

18. The computer program tool as recited in claim 14, wherein the converting program code further comprises program code for wrapping the read only file with executable code.

19. The computer program tool as recited in claim 15, wherein the read only file is an image file.

20. The computer program tool as recited in claim 15, wherein the read only file is an audio file.

21. The method as recited in claim 6, wherein the wrapping step further comprises the step of wrapping the read only file with code headers and records in order to appear to an operating system loader as the executable code.

22. The data processing system as recited in claim 12, wherein the wrapping circuitry wraps the read only file with code headers and records in order to cause an operating system loader to treat the read only file as the executable code.

23. The computer program tool as recited in claim 18, wherein the program code for wrapping wraps the read only file with code headers and records so that this program code now is treated by an operating system loader as the executable code.

24. The data processing system as recited in claim 7 wherein the system includes a processor operating in a protected mode.

25. The method as recited in claim 2, wherein the processor operates in a protected mode.

26. The computer program tool as recited in claim 14, wherein the processor operates in a protected mode.

27. A method comprising the steps of:  
converting a read-only file into a shared library with an executable code segment, wherein the shared library includes a reference pointer to data residing in the executable code segment;  
an application that needs to use the data referencing a pointer to the data; and  
an operating system faulting a page containing a memory segment wherein the read-only file resides into memory.